Redefining lobe-wise ground-glass opacity in COVID-19 through deep learning and its correlation with biochemical parameters -Supplementary Text-

I. Step-by-Step Demonstration

This section details how the entire Deep Learning (DL) and Ground Glass Opacity (GGO) detection is carried out step by step.

A. Pre-training the Deep-Unet Model for transfer learning

Firstly, for transfer learning LUNA-16 dataset was used.

1) Step 1: Format Synchronisation: The 3D Masks and the 3D image files are in varied format (MHD/RAW and NRRD format, respectively). Both of them are converted into the NIFTI format using a python script.

2) Step 2: Converting 3D NIFTI Images into 2D Image slices: Our Target Dataset is a collection of segmented 2D PNG images, thus for transfer learning we need to convert 3D NIFTI (Fig. 1) into 2D PNG images (Fig. 2) of same sizes. The orientation of the target dataset images is important to note as the final slices from 3D will have to be of the same orientation. Another important note is that the target dataset contains all lobes, but that is not the case for all 3D to 2D image slices. The 2D Slices with a missing lobe are rejected. This is done using a python script. The same procedure was followed with the mask files (Fig. 3).

3) Step 3: Segregation into training, testing and validation sets: The extracted 2D images were divided into training, testing and validation sets in the ratio 80:10:10.

4) Step 4: Augmentation: The following augmentations were randomly applied to the 2D images in order to avoid overfitting: Horizontal Flip, Shift-Scale-Rotate, Random Crop, Additive Gaussian Noise, Perspective, Brightness, Gamma, Sharpen, Blur, Contrast, Hue-Saturation.

5) Step 5: Training: The Deep-Unet model is initialised, and trained on the LUNA-16 dataset. The final model was saved to disk for further training.

B. Training the model on the target dataset.

Now that our model is pre-trained on LUNA-16 dataset, we can proceed with training our own dataset.

1) Step 1: Preprocessing: A similar pipeline was followed to the LUNA-16 dataset (after the extraction of 2D slices) for preprocessing.

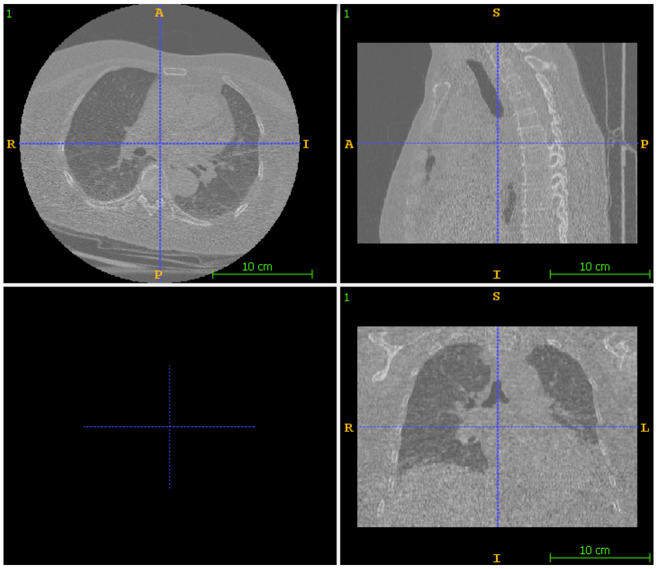

Fig. 1. A 3D NIFTI Image

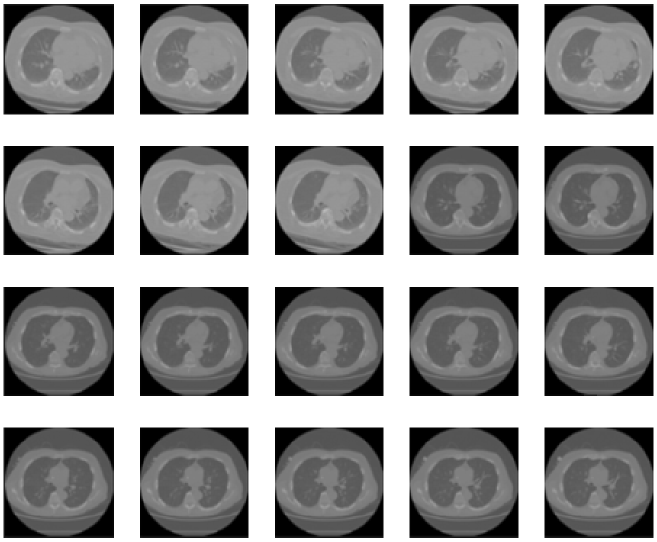

Fig. 2. A collection of 2D Lung MRI Slices

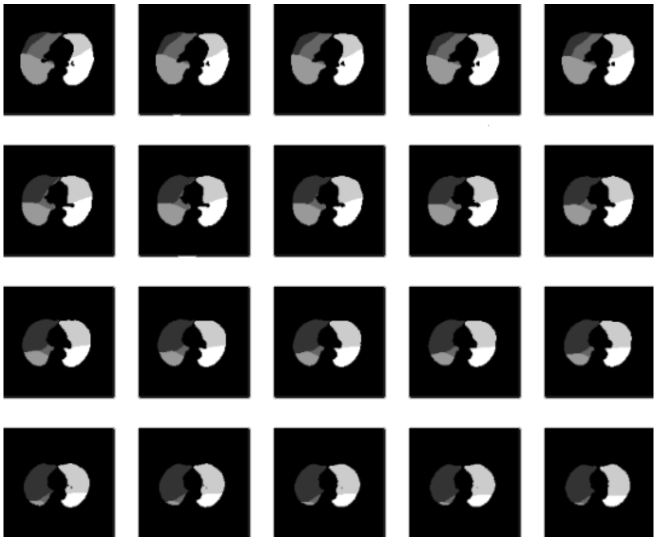

Fig. 3. A collection of 2D Mask Slices

2) Step 2: Training: The pre-trained Deep-Unet model is loaded, and trained on the target dataset. The final model was saved to disk for further evaluations.

C. GGO Detection

In this section, we show the step-by-step output of the post-processing steps.

1) Figure 4 shows a CT scan of a randomly chosen subject.
2) The lobes after segmentation are passed through a vessel removal process. Figure 5 shows the output received after vessels removal, wherein, we can observe that both the large and small vessels from each of the lobes have been removed.
3) Then, we further remove some remaining small regions from the lobes and perform clustering of pixels. The output after this stage is shown in figure 6.

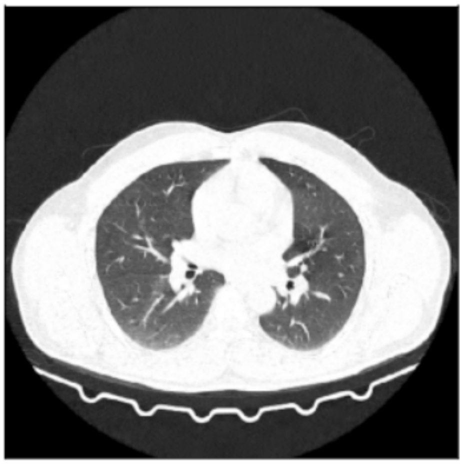

Fig. 4. A 2D CT scan belonging to a subject chosen randomly

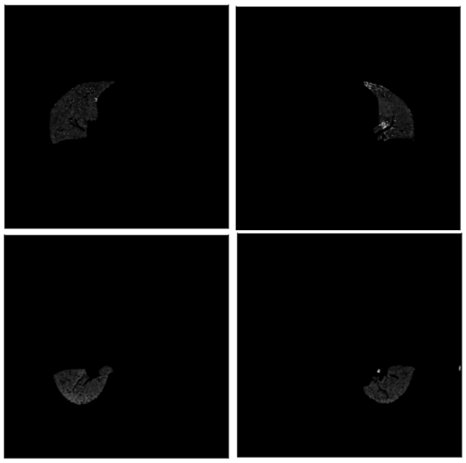

Fig. 5. The four lobes after performing vessel removal. Top left: Lobes 1 and 2; Bottom left: Lobe 3, Top right: Lobe 4; Bottom right: Lobe 5.

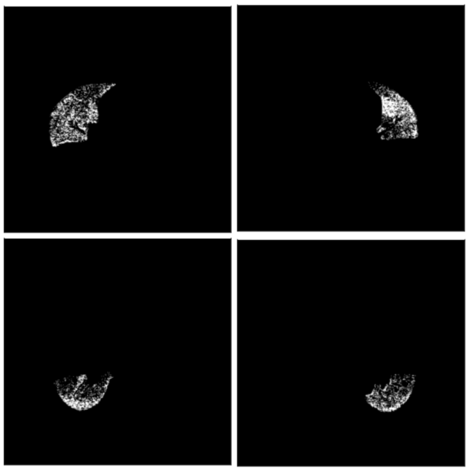

Fig. 6. The four lobes after performing removing small regions and clustering. Top left: Lobes 1,2; Bottom left: Lobe 3, Top right: Lobe 4; Bottom right: Lobe 5.

References

[1] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[5] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-first AAAI conference on artificial intelligence*, 2017.

TABLE I

IOU Scores for various backbones, calculated at an early stage with a smaller dataset and 40 epochs.

	efficientnetb3 [1]	vgg19 [2]	resnet152 [3]	resnext101 [4]	seresnext101 [5]	densenet201 [6]	inceptionresnetv2 [7]	efficientnetb7 [1]
Training	0.725435	0.389458	0.723235	0.727936	0.757175	**0.79518**	0.755838	0.703488
Validation	0.696909	0.345058	**0.748383**	0.695357	0.729264	0.731001	0.696584	0.711415
Testing	0.643457	0.377448	0.658859	0.716402	0.708146	**0.721159**	0.683281	0.686485

TABLE II

F1 Scores for various backbones, calculated at an early stage with a smaller dataset and 40 epochs.

	efficientnetb3 [1]	vgg19 [2]	resnet152 [3]	resnext101 [4]	seresnext101 [5]	densenet201 [6]	inceptionresnetv2 [7]	efficientnetb7 [1]
Training	0.808266	0.481535	0.81988	0.808551	0.820208	**0.874597**	0.825369	0.7678
Validation	0.748317	0.426726	**0.793801**	0.749509	0.783249	0.781655	0.742178	0.766961
Testing	0.702199	0.441163	0.740007	0.774276	0.768251	**0.778462**	0.752834	0.741006